Implementation Process

The Roman Urdu Diaries Corpus was loaded from text files.The text was split into words (tokenized), and words with numbers were removed.

The training data was used to train unigram, bigram, and trigram models. A unigram model counted individual words, a bigram model counted two-word combinations, and a trigram model counted three-word combinations.

Sentences were created from the instructed models: Unigram, bigram, trigram, backward bigram, and bidirectional bigram models. The generatetext function generated sentences using the specified model.

Each model's prediction of the test data was evaluated with perplexity. A model with lower perplexity is more accurate in predicting the next word.

Challenges Faced

1. **Backward and Bidirectional Models**:Generating text backward or in both directions was tricky and required careful handling of word sequences.
2. **Corpus Size and Quality**:Model performance was dependent on the quality and quantity of the dataset. A larger dataset with higher quality would yield better performance.
3. **Computational Complexity**:Trigram models were more time-consuming to train and test than unigram and bigram models because of the additional context they used.

Comparison of Different Ngram Models

The perplexity results for the models are:

- **Unigram Perplexity**: 44.10
- **Bigram Perplexity**: 1.74
- **Trigram Perplexity**: 1.08

Conclusion

- The **unigram model** is simple but performs poorly, as shown by its high perplexity of 44.10.
- The **bigram model** is better, with a perplexity of 1.74, and generates more coherent sentences.
- The **trigram model** is the best, with the lowest perplexity of 1.08, and produces the most meaningful sentences.
- **Backward and bidirectional models** are interesting but need more work to improve their performance.